

ARTIFICIAL INTELLIGENCE (AI) SUSUN ANGKA BENTUK KOTAK 4 X 4 MENGGUNAKAN PENCARIAN HEURISTIK DENGAN ALGORITMA BFS

Lamria Manalu¹, Lamhot Sitorus²

¹Teknik Informatika Unika St. Thomas S.U; Jln. Setia Budi No.479-F Medan, 061-8210161

²Teknik Informatika Unika St. Thomas S.U; Jln. Setia Budi No.479-F Medan, 061-8210161

e-mail: ¹manalulamria@gmail.com, ²lamhot68@yahoo.com

Abstrak

Dalam kehidupan sehari-hari maupun dalam literatur-literatur dapat ditemukan berbagai jenis permainan angka. Keunikan dari permainan-permainan angka ini menjadikan permainan ini sangat mengasyikkan, dan sekaligus dapat digunakan untuk melatih kecerdasan. Permainan pergeseran angka biasanya dimainkan dalam bintang berkaki lima atau berkaki enam. Jenis permainan ini cenderung lebih mudah untuk dimainkan dan diselesaikan. Permainan ini akan menjadi jauh lebih rumit dan sukar apabila dimainkan dalam wadah yang berbentuk kotak. Bentuk wadah ini menyebabkan arah proses pergeseran angka menjadi terbatas.

Permainan pergeseran angka dalam kotak ini sangat rumit dan sukar untuk diselesaikan secara manual. Oleh karena itu, penulis berusaha untuk merancang sebuah perangkat lunak yang dapat mencari solusi terpendek bagi permainan ini dengan menggunakan bantuan pohon pelacakan.

Dengan alasan diatas maka penulis merancang suatu perangkat lunak yang dapat mencari solusi terpendek dari permainan pergeseran angka dalam kotak. Bentuk kotak 4 x4 , dengan jumlah titik (node) dalam kotak sebanyak 16 buah, dan jumlah titik yang dapat diisi adalah sebanyak 15 buah.

Kata kunci : Visual Studio, breadth-first search (BFS), algoritma heuristik, Perangkat lunak.

Abstract

In everyday life as well as in the literature can be found various types of numbers game. The uniqueness of these number games makes this game very exciting, and at the same time can be used to train intelligence. The numbers shift game is usually played in a five-legged or six-legged star. This type of game tends to be easier to play and complete. This game will become much more complicated and difficult when played in a box-shaped container. The shape of this container causes the direction of the process of shifting the numbers to be limited.

The game of number shifting in this box is very complicated and difficult to complete manually. Therefore, the author seeks to design a software that can find the shortest solution for this game using the help of a tracking tree.

With the above reasons, the authors designed a software that can find the shortest solution of the game shifting numbers in the box. Box 4 x4 shape, with the number of points (nodes) in the box as many as 16 pieces, and the number of points that can be filled is as many as 15 pieces.

Keywords : Visual Studio, breadth-first search (BFS), heuristic algorithm, Software.

1. PENDAHULUAN

dalam kehidupan sehari-hari maupun dalam literatur-literatur dapat ditemukan berbagai jenis permainan angka. Keunikan dari permainan-permainan angka ini menjadikan permainan ini sangat mengasyikkan, dan sekaligus dapat digunakan untuk melatih kecerdasan. Permainan pergeseran angka biasanya dimainkan dalam bintang berkaki lima atau berkaki enam. Jenis permainan ini cenderung lebih mudah untuk dimainkan dan diselesaikan. Permainan ini akan menjadi jauh lebih rumit dan sukar apabila dimainkan dalam wadah yang berbentuk kotak. Bentuk wadah ini menyebabkan arah proses pergeseran angka menjadi terbatas.

Permainan pergeseran angka dalam kotak ini dapat diselesaikan dengan menggunakan bantuan struktur pohon pelacakan (*search tree*). Pohon pelacakan adalah suatu pohon (*tree*), dimana akar dari pohon berupa keadaan awal dan cabang berupa keadaan-keadaan yang mungkin terjadi dari keadaan sebelumnya serta daun merupakan keadaan akhir, yang dapat dijadikan sebagai solusi dari permasalahan. Namun, tidak semua daun dapat dijadikan sebagai solusi, dalam beberapa contoh kasus, ada beberapa atau semua daun bukan merupakan solusi dari permasalahan. Algoritma pencarian yang akan digunakan adalah pencarian heuristik dengan algoritma *breadth-first search* (BFS). Pada algoritma BFS, pencarian dimulai dari *node* akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Pada pencarian heuristik, pencarian dibimbing menuju *goal state* dengan pengetahuan-pengetahuan yang telah dihasilkan sebelumnya.

2. METODE PENELITIAN

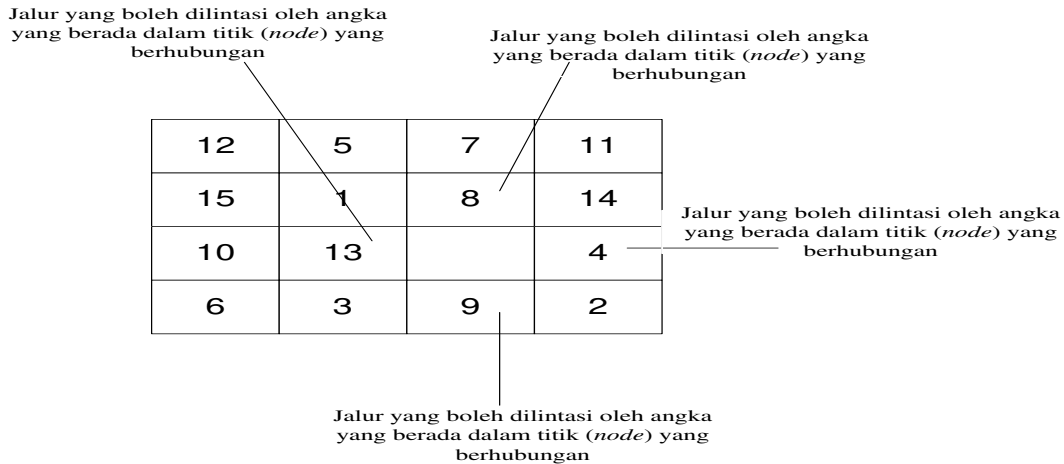
2.1 Analisis

Dalam suatu pembangunan aplikasi, analisis perlu dilakukan sebelum tahap perancangan dilakukan. Perancang aplikasi harus menganalisis kebutuhan apa saja yang diperlukan untuk membangun suatu perangkat lunak. Sebelum membangun aplikasi diperlukan identifikasi awal, yaitu:

1. Aplikasi yang akan dibuat adalah sebuah permainan (*game*), yaitu permainan pergeseran angka bentuk 4 x 4 dengan menggunakan konsep kecerdasan buatan dengan algoritma *Breadth First Search*.
2. Pengguna dari aplikasi ini tidak terbatas usia. Karena aplikasi ini merupakan sebuah permainan maka bisa digunakan oleh anak-anak dan orang dewasa.
3. Aktifitas yang akan dijumpai dalam permainan ini adalah pemain ditentang berpikir kreatif bagaimana untuk membuat semua angka terletak pada posisi sebenarnya dengan mengikuti aturan pergeseran yang telah ditetapkan.

2.1.1 Analisis Permainan Pergeseran Angka Bentuk Kotak 4 x 4

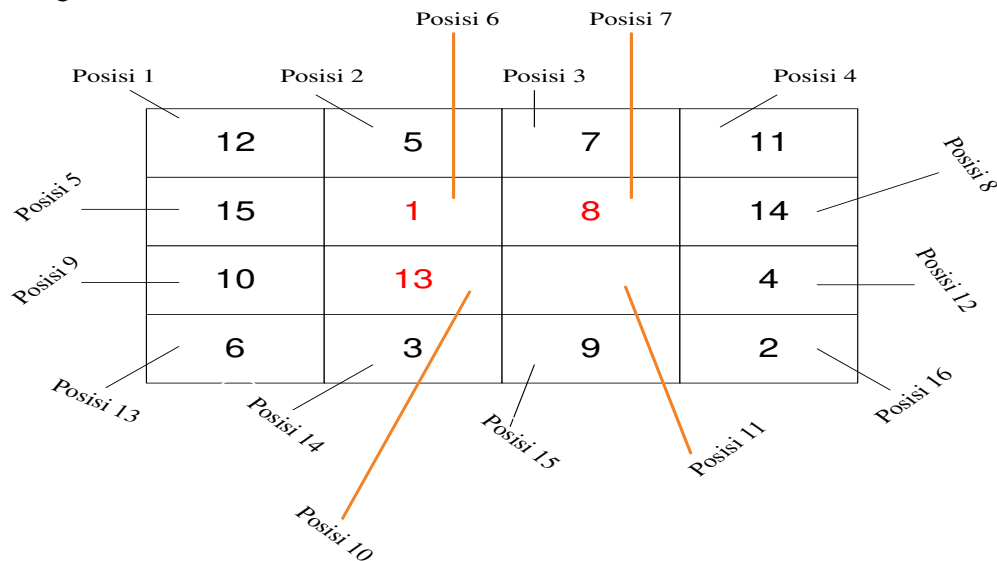
Setiap pergeseran angka harus mematuhi aturan yang telah ditetapkan. Pada kotak 4 x 4 yang memiliki 15 buah titik, tersedia 16 buah titik yang memuat angka dan 1 buah titik kosong. Aturan pergeseran angka adalah seperti pada Gambar 1.



Gambar 1 Aturan pergeseran angka dalam kotak 4 x 4

Pada Gambar 2 terdapat jalur yang menghubungkan antara titik yang berisi angka dengan titik yang kosong, sehingga angka 8, 4, 9 dan 13 bisa digeser ke titik yang kosong.

Setiap titik dalam kotak diberi nomor urut dari posisi baris paling atas, dari kiri ke kanan, turun ke bawah dan seterusnya. Selanjutnya, angka-angka disimpan dalam bentuk *string* berurut sesuai dengan posisi dari titik yang ditempatinya. Tempat kosong direpresentasikan sebagai angka '0'.



Gambar 2 Posisi titik pada bentuk Kotak 4 x 4

Kondisi kotak 4 x 4 pada Gambar 2 dapat disimpan dalam bentuk *string*, yaitu '12-5-7-11-15-1-8-14-10-13-0-4-6-3-9-2'. Pergeseran angka bentuk kotak 4 x 4 memiliki keadaan awal (*initial state*) dan keadaan tujuan (*goal state*) yang hendak dicapai.

2.1.2 Analisis Algoritma Breadth First Search pada Permainan Pergeseran Angka Bentuk Kotak 4 x 4.

1. Algoritma pencarian melebar pertama BFS (*breadth first search*)

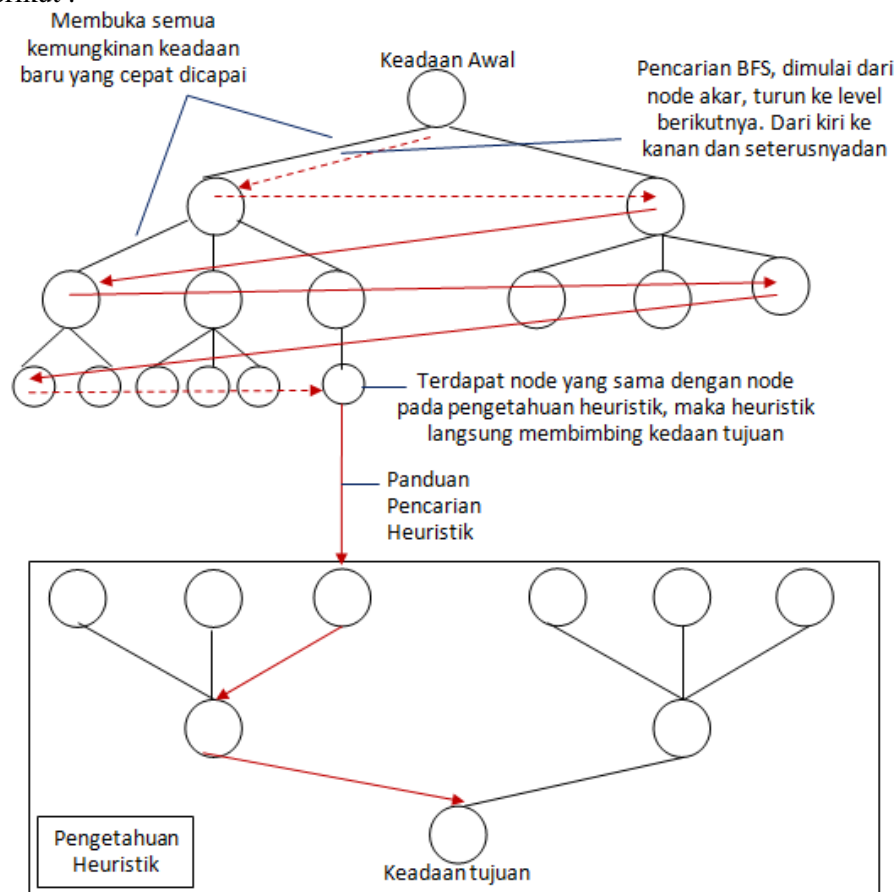
Pada metode pencarian BFS, semua *node* pada level *n* akan dikunjungi terlebih dahulu sebelum mengunjungi *node-node* pada level *n+1*. Pencarian dimulai dari *node* akar (keadaan awal) terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Pada implementasinya, algoritma BFS akan memerlukan waktu yang relatif lama untuk menemukan

solusi yang memiliki langkah penyelesaian yang relatif panjang, karena untuk mendapatkan solusi pada level n , BFS akan mengembangkan semua *node* pada level sebelum n .

2. Algoritma pencarian heuristik

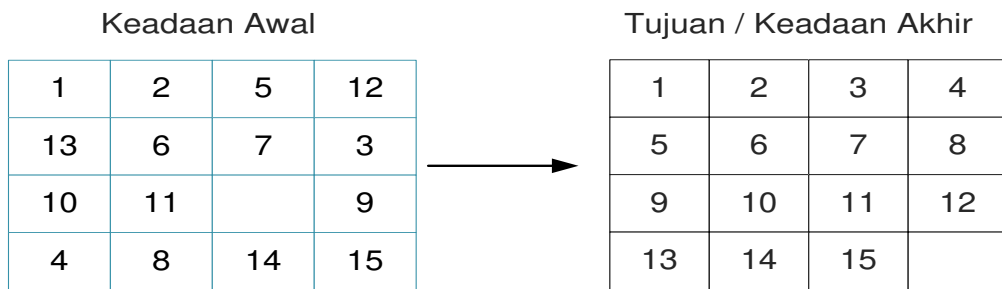
Algoritma pencarian heuristik memiliki pengetahuan untuk membimbing pencarian menuju ke keadaan tujuan. Pengetahuan ini berfungsi untuk membimbing arah pencarian ke keadaan tujuan. Pengetahuan yang dimaksud adalah *list* keadaan mulai dari keadaan tujuan, turun ke level bawahnya (seperti prosedur pengembangan *node* baru pada pohon pelacakan) dan seterusnya. Semakin besar *level node*, maka semakin besar pengetahuan, semakin jauh jangkauan bimbingan dan semakin cepat keadaan tujuan ditemukan. Pada perancangan perangkat lunak, algoritma pencarian heuristik dikombinasikan dengan pencarian BFS. Posisi pencarian heuristik adalah sebagai pembantu atau pembimbing bagi metode BFS. Apabila *node* yang dikembangkan oleh BFS terdapat dalam *list* pengetahuan heuristik, maka pencarian akan langsung dibimbing menuju keadaan tujuan sesuai dengan pengetahuan yang dimiliki pencarian heuristik.

Gambaran pencarian BFS dan bantuan pencarian heuristik dapat dilihat pada gambar sebagai berikut :



Gambar 3 Gambaran proses kerja BFS dan heuristik dalam mencari solusi

Seperti yang telah dipaparkan pada bab sebelumnya, aplikasi yang dibangun adalah aplikasi yang akan mencari solusi terpendek pada permainan pergeseran angka bentuk kotak 4 x 4 dengan menggunakan algoritma *Breadth First Search*. Algoritma ini akan diterapkan pada menu 'solve' pada aplikasi. Menu 'solve' adalah menu untuk mencari solusi pada permainan ini.



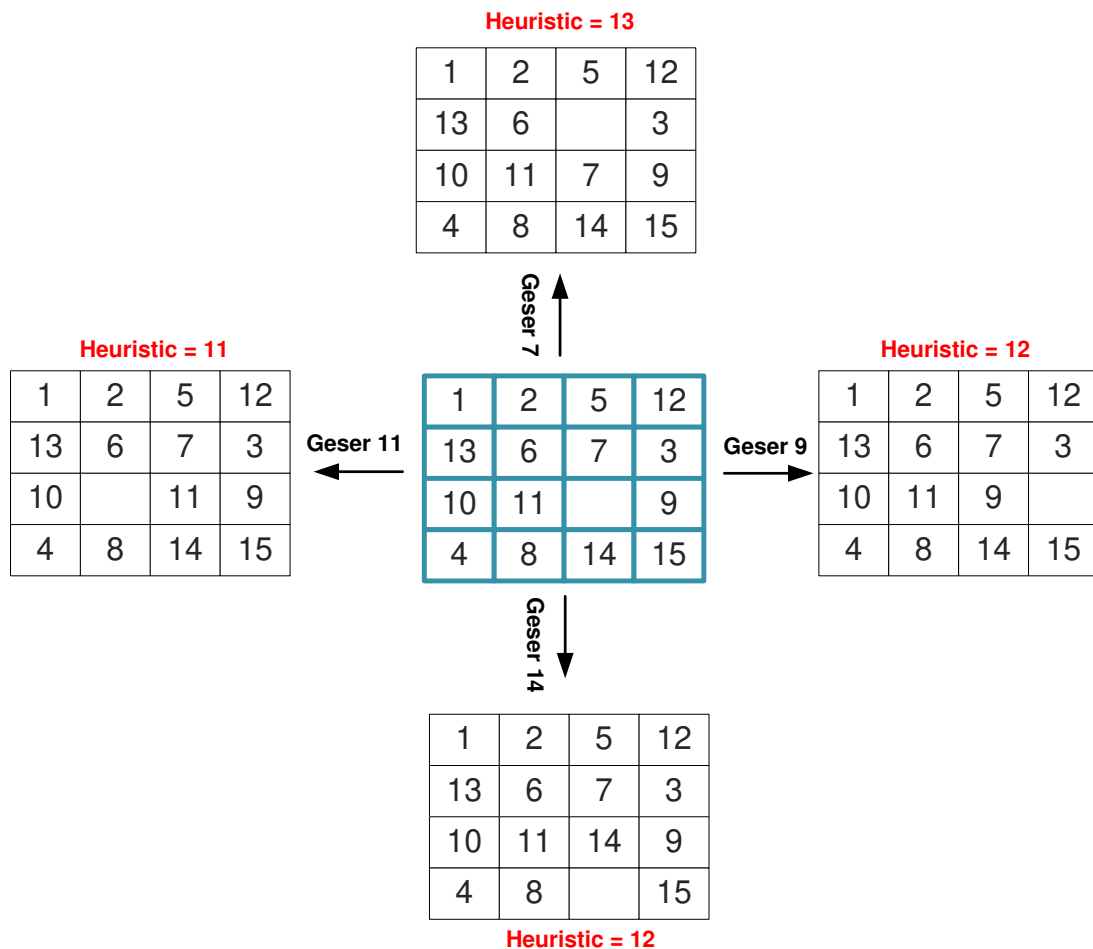
Gambar 4 Contoh keadaan awal dan keadaan akhir yang ingin dicapai pada permainan pergeseran angka bentuk 4 x 4

Contoh permasalahan pada Gambar III.4 jika diurutkan sesuai urutan posisi yang sudah ditetapkan sebelumnya yaitu keadaan awal '1-2-5-12-13-6-7-3-10-11-0-9-4-8-14-15' dan tujuan/keadaan akhir '1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-0'.

Dengan menggunakan bentuk pohon untuk merepresentasikan ruang keadaan, berikut akan digunakan algoritma *Breadth First Search* untuk mencari langkah-langkah yang harus ditempuh dari keadaan awal sampai mendapatkan tujuan:

$h1$ = jumlah angka yang menempati posisi yang salah (*misplaced tiles*), maka jumlah yang lebih kecil adalah yang diharapkan (lebih baik). Struktur pohon pencarian dapat dilihat pada Gambar 5.

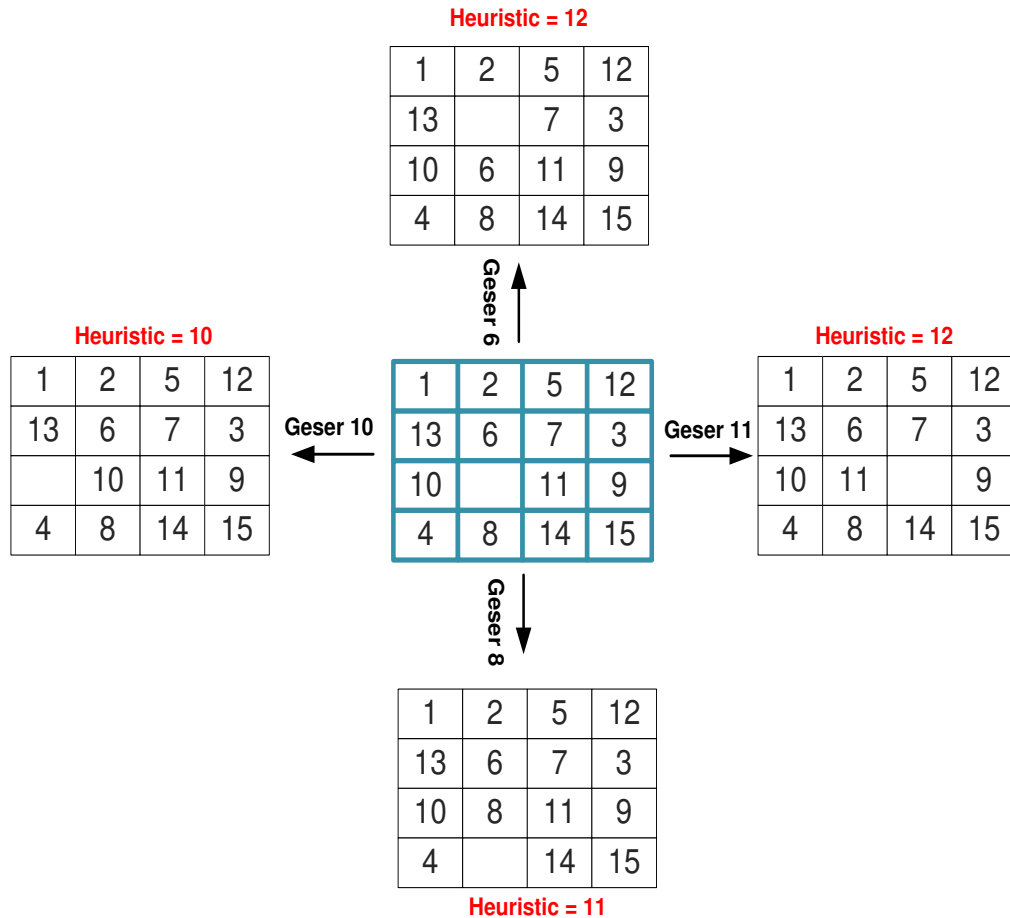
Langkah 1



Gambar 5 Pohon pencarian *Breadth First Search* pada kotak 4 x 4 Langkah 1

Berdasarkan gambar di atas, pertama kali dibangkitkan *node* ‘1-2-5-12-13-6-7-3-10-11-0-9-4-8-14-15’. Kemudian semua *successor* ‘1-2-5-12-13-6-7-3-10-11-0-9-4-8-14-15’ dibangkitkan, dan dicari nilai heuristik paling minimal. Pada pergeseran angka 11, *node* ‘1-2-5-12-13-6-7-3-10-0-11-9-4-8-14-15’ terpilih karena nilai heuristiknya paling rendah, yakni 11.

Langkah 2



Dan seterusnya

Gambar 6 Pohon pencarian *Breadth First Search* pada kotak 4 x 4 Langkah 2

Berdasarkan gambar di atas, *node* “1-2-5-12-13-6-7-3-10-0-11-9-4-8-14-15” terpilih dari heuristik paling minimal dari gambar III.5. Kemudian semua *successor* ‘1-2-5-12-13-6-7-3-10-0-11-9-4-8-14-15’ dibangkitkan, dan dicari nilai heuristik paling minimal lagi. Pada pergeseran angka 1, *node* ‘1-2-5-12-13-6-7-3-0-10-11-9-4-8-14-15’ terpilih karena nilai heuristiknya paling rendah, yakni 10 dan proses selanjutnya akan di ulang lagi sampai nilai heuristik mencapai 0

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Adapun hasil Perancangan Artificial Intelligence Susun Angka Bentuk Kotak 4 x 4 Menggunakan Algoritma BFS (Bread First Search) Dengan Heuristic.

3.1.1. Implementasi Sistem

Implementasi sistem adalah prosedur yang dilakukan untuk menyelesaikan desain yang ada dalam dokumen desain sistem yang disetujui dan menguji, menginstal, memulai, serta menggunakan sistem yang baru atau sistem yang diperbaiki.

Penggunaan suatu komputer untuk pemecahan masalah membutuhkan suatu sistem yang baik, sehingga memungkinkan berhasilnya komputer dalam melaksanakan tugasnya, yaitu mengolah data menjadi informasi. Langkah implementasi yang dilakukan dalam menyelesaikan Perancangan *Artificial Intelligence* Susun Angka Bentuk Kotak 4 x 4 Menggunakan Algoritma BFS (Breadth First Search) Dengan Heuristik, Menyediakan perangkat keras (Hardware) dan perangkat lunak (Software). Dalam tahap ini disediakan perangkat keras.

Perangkat lunak yang dibutuhkan adalah Sistem Operasi XP dan bahasa pemrograman yang digunakan untuk menulis program ke dalam komputer. Menguji sistem menjelaskan mengenai hasil pengujian sistem yang dilakukan pada Perancangan *Artificial Intelligence* Susun Angka Bentuk Kotak 4 x 4 Menggunakan Algoritma BFS (Breadth First Search) Dengan Heuristik. Metode pengujian sistem yang digunakan adalah *black-box testing*. *Black-box testing* adalah metode pengujian yang dimana penilaian terhadap sebuah aplikasi bukan terletak pada spesifikasi logika/fungsi aplikasi tersebut, tapi masukan (*input*) dan keluaran (*output*). Dengan berbagai masukan (*input*) yang diberikan akan dievaluasi apakah suatu sistem/aplikasi dapat memberikan keluaran (*output*) yang sesuai dengan harapan penguji. Pengujian sistem dilakukan dengan cara sebagai berikut:

- Hasil pengujian sistem disajikan dalam bentuk tabel.
- Pengujian ditargetkan pada setiap proses yang dimiliki Perancangan *Artificial Intelligence* Susun Angka Bentuk Kotak 4 x 4 Menggunakan Algoritma BFS (Breadth First Search) Dengan Heuristik.

3.1.2. Pembahasan Hasil

a. Form Splash Screen

Sewaktu menjalankan program form splash screen akan muncul pertama sekali sebagai loading menuju form utama, pada form splash screen ini informasi judul skripsi dan nama mahasiswa akan di tampilkan, dalam beberapa detik form splash screen ini muncul akan langsung menuju form utama, dan form splash screen akan di close.

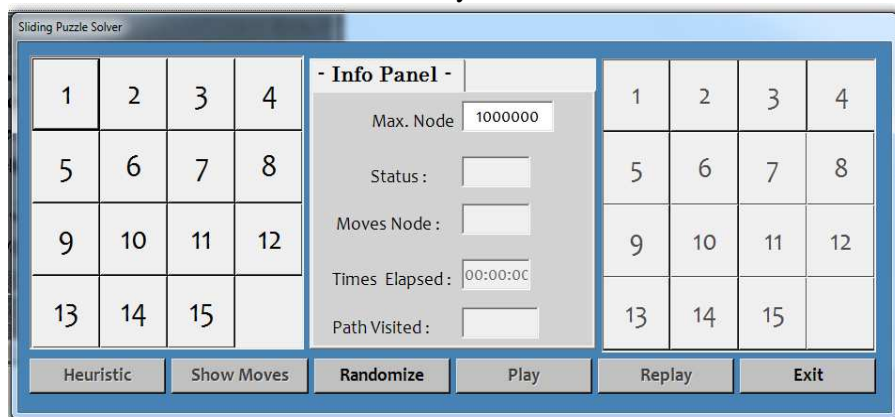


Gambar 7 Form Splash Screen

b. Form Utama

Form utama ini akan tampil setelah form splash screen selesai di loading. Pada form utama ini terdapat 2 susunan angka, susunan angka sebelah kiri untuk pencarian otomatis, susunan angka sebelah kanan penyusunan angka secara manual, juga terdapat beberapa tombol button, dimana fungsi – fungsi dari tombol ini adalah sebagai berikut :

1. Heuristic : Jika tombol ini di click akan mencari solusi node yang dilalui untuk menyusun angka : 1.2.4.5.6.7.8.9.10.11.12.13.14.15
2. Show Moves : Tombol ini aktif jika solusi node ditemukan pada tombol Heuristic, Jika tombol ini di click akan menggerakkan node per node menuju susunan : 1.2.4.5.6.7.8.9.10.11.12.13.14.15
3. Randomize : Jika tombol ini di click akan mengacak angka
4. Play : Jika tombol ini di click akan memulai permainan puzzle secara manual
5. Replay : Jika tombol ini di click mengembalikan susunan puzzle secara manual yang tidak selesai
6. Exit : Keluar dari system.

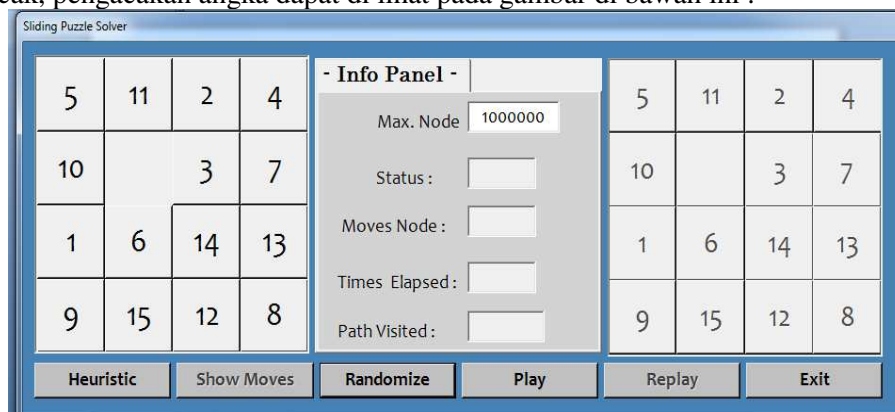


Gambar 8 Form Utama

c. Uji Coba Program 1

1. Click tombol Randomize :

Pada form ini akan di uji coba program puzzle 4 x 4 penyusunan angka, tombol yang di click pertama sekali adalah tombol “Randomize”. Maka keadaan angka akan di acak, pengacakan angka dapat di lihat pada gambar di bawah ini :



Gambar 9 Pengacakan Angka/Node 1

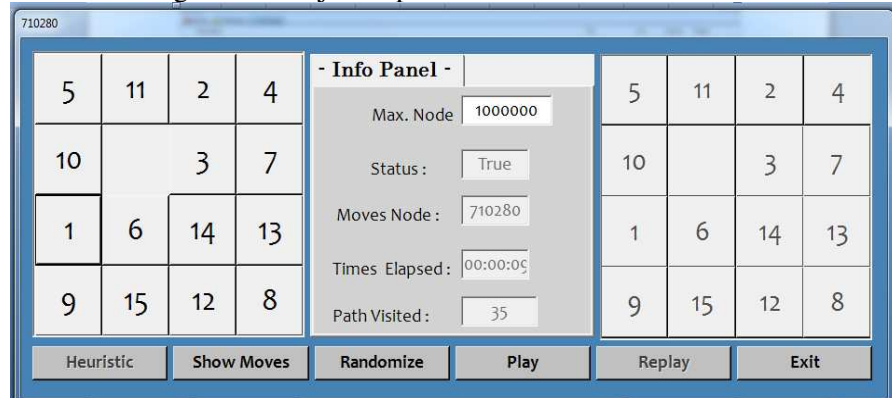
Pada pengacakan angka pertama ini, dapat dilihat keterangan-keterangan sebagai berikut pada panel [info panel] :

- a. Max Node = Maksimal node yang di lalui defaultnya 1000000 node, maksudnya jika node yang dilalui lebih besar dari 1000000 node maka solusi tidak di temukan

- b. Status = Jika Node yang di lalui lebih kecil dari 1000000 maka status = true, berarti solusi ditemukan, jika lewat dari 1000000 node maka status = false
- c. Moves Node = Node yang di lalui untuk penyusunan angka puzzle 4x4
- d. Times Elapsed = Waktu yang di butuhkan untuk mencari solusi
- e. Path Visited = Jumlah node yang di gerakkan jika solusi di temukan.

2. Click tombol Heuristik :

Pada kasus di atas tombol Heuristik di click untuk mencari solusi penyusunan puzzle 4 x 4, maka gambar menjadi seperti di bawah ini :



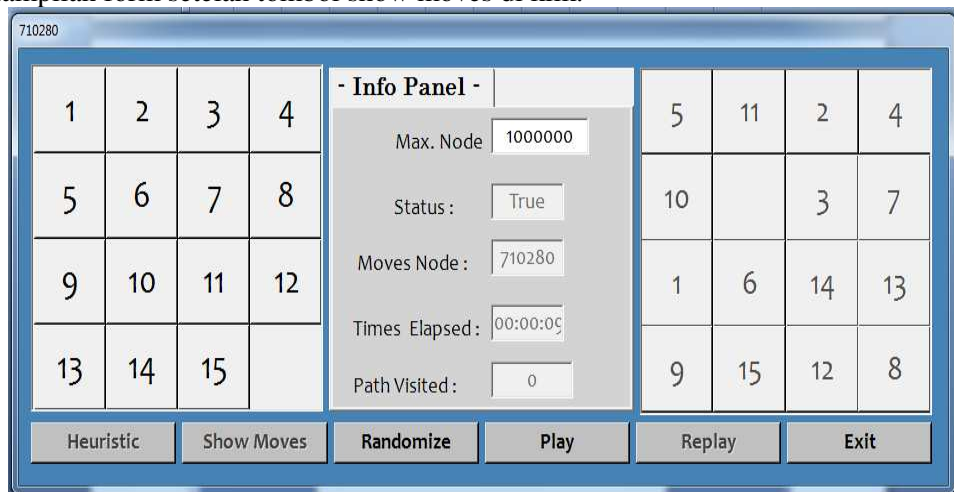
Gambar 10 Pencarian Solusi Heuristik

Setelah tombol heuristik di klik maka info panel berubah menjadi seperti di berikut :

- a. Status menjadi true menandakan solusi di temukan kurang dari max node 1000000
- b. Mode node yang di lalui sebanyak 710280 node
- c. Times Elapsed yang di perlukan untuk mencari solusi sebesar 00.00.09.
- d. Path Visited sebesar 35 node yang di gerakkan.

3. Click tombol Show moves :

Karena solusi di temukan maka tombol show moves akan aktif, dan jika tombol show moves di klik maka angka akan bergeser sendiri setiap detik sebanyak path visited akan di hitung mundur sampai 0, pada kasus ini path visited sebesar 35 node. Berikut tampilan form setelah tombol show moves di klik.

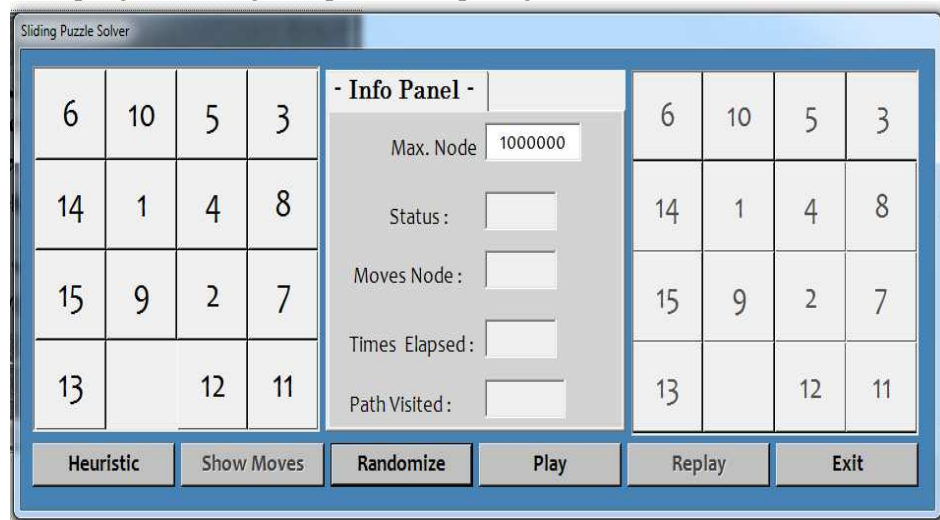


Gambar 11 Show Moves

d. Uji Coba Program 2

1. Click tombol Randomize :

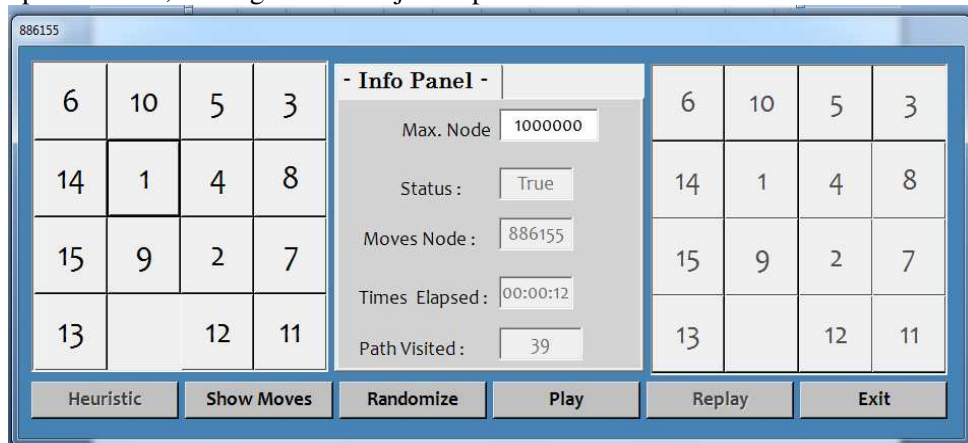
Pada form ini akan di uji coba program puzzle 4 x 4 penyusunan angka, tombol yang di click pertama sekali adalah tombol “Randomize”. Maka keadaan angka akan di acak, pengacakan angka dapat di lihat pada gambar di bawah ini :



Gambar 12 Pengacakan Angka/Node 2

2. Click tombol Heuristik :

Pada kasus di atas tombol Heuristik di click untuk mencari solusi penyusunan puzzle 4 x 4, maka gambar menjadi seperti di bawah ini :



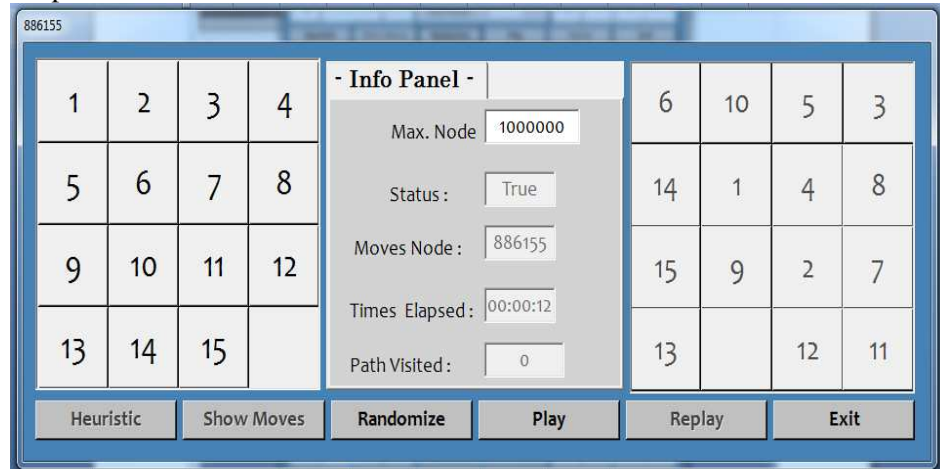
Gambar 13 Pencarian Solusi Heuristic

Setelah tombol heuristik di klik maka info panel berubah menjadi seperti di berikut :

1. Status menjadi true menandakan solusi di temukan kurang dari max node 1000000
2. Mode node yang di lalui sebanyak 886155 node
3. Times Elapsed yang di perlukan untuk mencari solusi sebesar 00.00.12.
4. Path Visited sebesar 39 node yang di gerakkan.

3. Click tombol Show moves :

Karena solusi di temukan maka tombol show moves akan aktif, dan jika tombol show moves di klik maka angka akan bergeser sendiri setiap detik sebanyak path visited akan di hitung mundur sampai 0, pada kasus ini path visited sebesar 39 node. Berikut tampilan form setelah tombol show moves di klik.



Gambar 14 Show Moves

4. KESIMPULAN

Setelah menyelesaikan perangkat lunak penyelesaian permainan pergeseran angka dalam bentuk kotak, penulis menarik kesimpulan sebagai berikut:

1. Penggunaan metode BFS menjamin solusi yang ditemukan adalah solusi terpendek (*shortest path*).
2. Perangkat lunak merupakan implementasi nyata untuk penggunaan metode Algoritma *Breadth First Search* (BFS) dengan pencarian heuristik dalam mencari solusi pada suatu permasalahan berbasis *Artificial Intelligence* (AI).

5. SARAN

Penulis ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak ini yaitu:

1. Perangkat lunak dapat dikembangkan dengan menambahkan beberapa bentuk wadah lainnya, seperti: segi enam (heksagon), segi tujuh (heptagon), segi delapan (octagon) dan bentuk lainnya.
2. Perangkat lunak ini dapat dikembangkan dengan menambahkan metode pencarian lainnya, seperti: metode pencarian *Depth-First-Search*, *Hill-Climbing*, dan metode lainnya.
3. Perangkat lunak dapat dikembangkan dengan menambahkan penggambaran dan penjelasan pada pohon pelacakan yang digunakan untuk mencari solusi.

DAFTAR PUSTAKA

- [1] Desiani.A dan Arhami.M, 2006, *Konsep Kecerdasan Buatan*, Penerbit Graha Ilmu.
- [2] Kusumadewi.S, 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Edisi 2, Penerbit Graha Ilmu.
- [3] T.Sutojo, Edy Mulyanto, Dr. Vincent Suhartono, (2011), *Kecerdasan Buatan*, Penerbit Andi Yogyakarta.
- [4] Ramadhan. A, (2004), *36 Jam Belajar Komputer Visual Basic 2008*, PT. Elex Media Komputindo, Jakarta.
- [5] Supardi.Y, (2006), *Microsoft Visual Basic 2008 Untuk Segala Tingkat*, PT. Elex Media Komputindo, Jakarta.
- [6] http://eprints.undip.ac.id/5202/2/BAB_I_dan_II.pdf, di akses tanggal 24 Juni 2014.
- [7] <http://repository.usu.ac.id/bitstream/123456789/23413/4/Chapter%20II.pdf>, diakses tanggal 20 Juni 2014.
- [8] <http://hendrik.staff.gunadarma.ac.id/Downloads/files/23064/representasi%20pengetahuan.pdf>, diakses tanggal 18 Mei 2014.
- [9] <http://informatika.stei.itb.ac.id/rinaldi.munir/Stmik/2012,2013/Makalah2012/Makalah-IF3051-2012-037.PDF>, diakses tanggal 11 Juni 2014.